A Textual-Based Similarity Approach for Efficient and Scalable External Plagiarism Analysis

Lab Report for PAN at CLEF 2010

Daniel Micol, Óscar Ferrández, Fernando Llopis, and Rafael Muñoz

Research Group on Natural Language Processing and Information Systems
Department of Software and Computing Systems
University of Alicante
San Vicente del Raspeig, E-03080 Alicante, Spain
{dmicol, ofe, llopis, rafael}@dlsi.ua.es

Abstract In this paper we present an approach to detect external plagiarism based on textual similarity. This is an efficient and precise method that can be applied over large sets of documents. The system that we have developed contains a first phase of document selection that uses a variant of tf-idf applied over the terms that appear in the two documents of the pair being compared. After this is done, we apply a more complex and accurate function based on character n-grams over the subset of documents resulting from the first step in order to extract the plagiarized passages, or matches. Once all matches for a given document are extracted, we perform a greedy match merging operation to allow in-between text in order to be compatible with certain levels of plagiarism obfuscation. In our participation in the *2nd International Competition on Plagiarism Detection*, we achieved an overall score of 0.2222, ranking 11^{th} out of 18 participants.

1 Introduction

External plagiarism analysis is a complex task that attempts to determine if a suspicious document contains one or more appropriations of another text which belongs to a set of source candidates. It can be a very expensive task if the number of documents to compare against is large. This is the case for the corpora provided by the organizers of the *1st and 2nd International Competitions on Plagiarism Detection* [11,12], which we have used as training and evaluation sets for our participation in the latter competition. These corpora contain two sets of documents, namely suspicious and source. The first of them contains those documents that may include one or more plagiarisms extracted from documents from the second of these sets. In total, the corpora provided for this competition contain several thousands of documents of both kinds.

Given the large amount of data to process, and the number of document comparisons to perform, one of the main goals for the systems that participate in these competitions is to be highly efficient. For this purpose, we decided to apply a lightweight document similarity function that would be used as heuristic to determine if a given suspicious and source documents are similar enough to hold a plagiarism relation. After a set of candidate source documents is extracted, we apply a more expensive and accurate

function to detect the corresponding plagiarized fragments. This two-step architecture is in line with the current state of the art systems.

The remainder of this paper is structured as follows. The next section will describe the state of the art in external plagiarism analysis. The third will describe the methods implemented in our system, while the fourth one contains the experimental results of our approach. Finally, the fifth and last section presents our conclusions and proposes future work based on our current research.

2 State of the art

Most of the research approaches in the field of external plagiarism analysis contain a simple and efficient heuristic retrieval to reduce the number of source documents to compare every suspicious text against, and a more complex and costly detailed analysis that attempts to extract the exact position of the plagiarized fragment, if any [11]. As mentioned before, the system that we have developed is in line with this two-step architecture.

2.1 Heuristic retrieval

Given the large amount of data to process, and the number of document comparisons to perform, one of the main goals of the systems that participate in the previously mentioned competitions is to be highly efficient. Theoretically, every source document could be the origin of a given plagiarism from a suspicious document, so we need to compare all of them with each other. This has a squared complexity, assuming the number of documents in both sets is of the same magnitude, and therefore presents performance and scalability limitations. Because of this, performing a lightweight heuristic retrieval that reduces the number of comparisons to perform is highly recommendable.

Some authors, like [6], create an inverted index of the corpus documents' contents in order to be able to retrieve efficiently a set of texts that contain a given n-gram. Other authors, such as [3,5], decided to apply a document similarity function that would be used as heuristic to determine if a given suspicious and source documents are similar enough to hold a plagiarism relation.

On the other hand, some authors, like [5], implement a character-level n-gram comparison and apply a cosine similarity based on term frequency weights. With this approach they extract the 51 most similar source documents to the suspicious one being analyzed. Other authors, such as [2,6], decided to implement a word-level n-gram comparison. The first of them, [2], keeps the 10 most similar using a custom distance function based on frequency weights, while the second, [6], keeps those source documents that share at last 20 word-grams of length 5. Low granularity word n-grams, with a size of 1, have been explored by [8], applying cosine similarity using frequency weights to extract the two most similar partitions for every sentence in a document, using the source documents' sentences as centroid. Finally, [13] has applied the well-known document fingerprinting approach described in [14] with 50 character chunks and overlap of 30. The candidate source documents will be those that share at least one value with the suspicious document's fingerprint.

2.2 Detailed analysis

Regarding the detailed analysis, [5] extracts character-level n-grams, and later on performs a computation of the distances of adjacent matches, joining them based on a Monte Carlo optimization. Afterwards, they propose a refinement of the obtained section pairs. [6] extracts matches of word n-grams of length 5, and applies a match merging heuristic to obtain larger matches. Then they extract the maximal size which share at least 20 matches, including the first and the last n-gram of the matching sections, and for which 2 adjacent matches are at most 49 not-matching n-grams apart. On the other hand, [2] performs a greedy match merging if the distance of the matches is not too high.

A more strict approach has been presented by [8], requiring exact sentence matches, and afterwards applying a match merging approach by greedily joining consecutive sentences. In this method, gaps are allowed if the respective sentences are similar to the corresponding sentences in the other document. Finally, [13] requires an exact match of document fingerprints, doing an extraction of the pairs of sections which are obtained by enlarging matches and joining adjacent matches. Gaps are required to be below a certain Levenshtein edit distance.

3 Methods

As mentioned before, the corpora that we want to use our system against contain several thousands suspicious and source documents, where every suspicious may include plagiarisms of one or more source documents. If we were to compare all of them with each other, we would have to do millions of document comparisons. Given that every document can contain thousands or millions of characters, this set of comparisons becomes intractable.

To make this problem computationally feasible, we must reduce by a large factor the number of document comparisons to perform, by generating a subset of candidate source documents for every suspicious text. After this subset has been computed, we will be able to apply a more complex function that will detect the plagiarized fragments, if any. This measure should be as independent as possible of the document sizes, given that otherwise our system will not scale.

3.1 Document selection

As mentioned before, the first step would be to select a subset of candidate source documents that will later on be compared against a given suspicious document. This should reduce by a large factor the number of document comparisons to perform. To generate this set we will have to loop through all source documents, and given that this set is large, this operation needs to be efficient.

Our approach to solve this problem is to weight the words in every document and then compare the weights of those terms that appear in both the suspicious and the source documents being compared. The similarity score between the aforementioned two documents will be the sum of the common term weights.

Similarity measure To measure the similarity between a suspicious and a source document, we will use a variant of the *term frequency inverse document frequency* function [15], or tf-idf, commonly used in information retrieval. It is applied over a term, t_i, in a given document, d_j, and is defined as follows:

$$tf\text{-}idf_{i,j} = tf_{i,j} \cdot idf_i \tag{1}$$

where the term frequency of t_i in document d_j, or $tf_{i,j}$, is:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2}$$

being $n_{i,j}$ the number of occurrences of the considered term, t_i, in document d_j, and the denominator is the sum of the number of occurrences of all terms in document d_j.

Furthermore, the inverse document frequency of term t_i, or idf_i, is defined as:

$$idf_i = \log\left(\frac{|D|}{|\{d : t_i \in d\}|}\right) \tag{3}$$

where $|D|$ is the number of documents in our corpus.

For our purpose this definition of tf-idf is not optimal. The tf value is normalized by the length of a document to prevent longer documents from having a higher weight. This makes sense in information retrieval applications such as search engines. However, in our case it would be better to skip this normalization given that the more times a word appears in a document, the higher likelihood it will have to contain a plagiarized fragment, regardless of its size. This is because there will be more passages that are similar and that therefore could be plagiarized. Therefore, tf in our case will be defined as:

$$tf_{i,j} = n_{i,j} \tag{4}$$

With regards to idf, given that in our case the number of documents in the corpus is constant and we will only need the relative tf-idf value, not the absolute, we can simplify this function by removing the $|D|$ and the logarithm:

$$idf_i = \frac{1}{|\{d : t_i \in d\}|} \tag{5}$$

Finally, the similarity score of two documents, being d_j the suspicious and d_k the source one, will be defined as shown in the following equation:

$$sim_{d_j,d_k} = \sum_{t_w \in d_j \cap d_k} (tf_{w,j} \cdot idf_{w,j}) = \sum_{t_w \in d_j \cap d_k} \left(n_{w,j} \cdot \frac{1}{|\{d : t_w \in d\}|}\right) \tag{6}$$

where $|d|$ represents the number of texts in the source documents corpus that contain a term that appears in both the suspicious and the source documents.

Therefore, the similarity of two documents will be higher the more words they have in common, and also the fewer documents those terms appear in.

Choosing the candidate source documents Now that the document similarity measure is defined, we describe the approach to compare suspicious and source documents. As mentioned before, the similarity measure will be applied over those terms that appear in both the corresponding suspicious and source documents being compared. For this purpose, we normalize and tokenize the contents of every document. This means that we only keep alphanumerical symbols, and the rest are replaced by spaces. After this process is done, we do a space-based tokenization to keep all tokens, filtering stop words and calculating the $tf\text{-}idf$ of all remaining terms.

When we compare two documents we will have two arrays with the $tf\text{-}idf$ of the terms in both of them, and will be able to calculate a similarity score as defined above. Each of these scores will be stored in a dictionary data structure where the key will be the corresponding term, and the value its $tf\text{-}idf$. Now comparing two documents is a linear complexity task.

For every suspicious document we will perform a comparison with all source documents in our corpus, and keep those that have the highest similarity scores. If we maintain a larger number of source documents we will have a higher recall for the next phase, but also higher response time, and vice versa for lower number of source documents.

3.2 Passage selection

Once we have a small set of source documents to compare against for every suspicious one, we can perform a more accurate and costly comparison between pairs of documents in order to detect the plagiarized fragments, or passages, if any.

Match extraction The approach that we have implemented first normalizes the contents of every document by keeping only alphanumerical characters, and removing spaces and punctuation symbols. This last point is a difference with the normalization mentioned before, and the reason for this is that we want our system to behave correctly against certain cases of low obfuscation like word-breaking or concatenation. Therefore, we use character-based n-grams over the normalized document contents. If we used word-based n-grams instead, our passage selection algorithm wouldn't behave well with the aforementioned obfuscation cases.

After the text has been normalized, we try to find the largest common substring between suspicious and source documents, requiring a minimum length which will be the n-gram size. Assuming that the suspicious document has n characters, and the source one has m, the complexity of a brute force solution for this problem would be $O(n \cdot m)$, which is of the order of $O(n^2)$. This is too expensive and presents scalability limitations, especially for large documents.

An optimization to this approach is to hash all overlapping n-grams of both the suspicious and source documents, storing, for every n-gram, the positions where they appear in the text. Once the n-grams of the source document being compared against have been hashed, we will iterate through the contents of the suspicious document, extract n-grams starting at every given offset, look them up in the hash of n-grams of the aforementioned source document, and go directly to the positions where the given n-gram appears, limiting unnecessary comparisons. From these points we will try to find

the largest common substring to both documents. A similar n-gram hashing technique is described in [14].

In our system we wanted to choose an n-gram size that would maintain a passage's meaning, and that is also not too strict. Smaller n-grams have the problem that they might match parts of a word and therefore the meaning would be lost. On the other hand, longer n-grams don't behave well when there is plagiarism obfuscation, so if the plagiarized sentence has been somehow altered, they won't work. In our experiments we used an n-gram size of 30 characters.

Match merging The result of the match extraction step will be a set of identical substrings that appear in both the suspicious and source documents. It would be beneficial to make this comparison less restrictive by allowing additional words or characters to appear in-between matches, so that our system works well on certain kinds of low level obfuscations. This is common among plagiarism when the corresponding person introduces additional words or rearranges part of the appropriated text. To overcome this issue, we perform a match merging operation, which attempts to group matches, and the text in-between, if they are close enough in both source and suspicious documents.

This is a greedy process that recursively attempts to merge matches if they satisfy the following two heuristics:

- The length of the text in-between two matches, m_i and m_j, is smaller than the length of m_i plus the length of m_j. This applies to the source and the suspicious part of the matches.
- The length of the merged match cannot be longer or equal than twice the length of m_i plus the length of m_j. This applies to the source and the suspicious part of the matches.

These two heuristics are similar to those presented in [2,6], and they have been obtained empirically.

In addition, at the end of the merging operation we will discard those matches that contain less than 100 characters in either source or suspicious document, as the minimum plagiarism size for the corpus that we use is 50 words [11], and therefore we believe that 100 characters is a safe lower limit.

4 Experimentation and results

We have performed two sets of experiments with our system. The first of them used an annotated training corpora to tune it, and the second one used an unannotated corpora in order to evaluate our approach.

The machine we used to carry out the experiments described in this section had 8 CPU cores, each of them running at a frequency of 2.53 GHz, and 16 GB of RAM. To maximize the usage of this machine's capabilities, we designed our system to be multi-threaded as most of the methods presented in this paper are fully parallelizable.

4.1 Training

To tune our system to behave optimally, we used the external plagiarism corpora from the *1st International Competition on Plagiarism Detection*, given that it contained annotations. This corpora contains a source documents corpus composed of $14,429$ texts, and a suspicious documents corpus composed of $14,428$ elements. Half of the suspicious texts contain a plagiarism, which ranges between 0% and 100% of the corresponding source document. In addition, the plagiarism length is evenly distributed between 50 and $5,000$ words.

The first aspect that we experimented with was trying to determine the optimal number of documents to be selected, given that a larger number of elements would lead to higher accuracy, but would affect performance negatively given that we would have more documents to process. The opposite applies to smaller selected document sets. Table 1 shows the results from this experiment using different set sizes, where column *Captured* represents the number of plagiarisms that are contained within the set of source documents, and *Missed* those that are not included in this set.

Table 1. Metrics using different selected document set sizes.

Size	Recall	Captured	Missed
1	0.3260	23,970	49,552
5	0.6875	50,547	22,975
10	0.7781	57,206	16,316
20	0.8282	60,893	12,629
30	0.8479	62,340	11,182
40	0.8595	63,189	10,333
50	0.8698	63,947	9,575
60	0.8760	64,403	9,119
70	0.8820	64,843	8,679
80	0.8869	65,205	8,317
90	0.8905	65,473	8,049
100	0.8941	65,734	7,788

Given the values shown in the previous table, we decided to use a number of documents of 10, since we believe it is the best trade-off between amount of texts and recall. After this step, we executed the passage selection, obtaining the following results: $overall = 0.3902$, $f\text{-}score = 0.5665$, $precision = 0.6873$, $recall = 0.4819$, and $granularity = 1.7354$. If we had participated in the *1st International Competition on Plagiarism Detection*, our system would have ranked 4^{th} out of 11 participants, as shown in Table 2. As we can see in this table, the strongest aspect of our system is its precision, where it ranks the third among all participants. On the other hand, recall and granularity were not as good, but still within the top half. The reason why recall is lower is in part due to the fact that we chose 10 source documents per suspicious text to evaluate, giving a maximum coverage value of 77.81%, as shown in Table 1. Apart from this, and since our method is purely textual, we miss plagiarisms that are not writ-

ten in similar ways. Finally, documents that are translated will also lower our recall. On the other hand, granularity would have been lower if we had been more aggressive at merging matches, although then precision might have suffered.

Table 2. Comparison of our system against the ones that participated in the *1st International Competition on Plagiarism Detection*.

Rank	Overall	F-score	Precision	Recall	Granularity	Participant
1	0.6957	0.6976	0.7418	0.6585	1.0038	[5]
2	0.6093	0.6192	0.5573	0.6967	1.0228	[6]
3	0.6041	0.6491	0.6727	0.6272	1.1060	[2]
4	**0.3902**	**0.5665**	**0.6873**	**0.4819**	**1.7354**	**This work**
5	0.3045	0.5286	0.6689	0.4370	2.3317	[9]
6	0.1885	0.4603	0.6051	0.3714	4.4354	[8]
7	0.1422	0.6190	0.7473	0.5284	19.4327	[13]
8	0.0649	0.1736	0.6552	0.1001	5.3966	[10]
9	0.0264	0.0265	0.0136	0.4586	1.0068	[16]
10	0.0187	0.0553	0.0290	0.6048	6.7780	[7]
11	0.0117	0.0226	0.3684	0.0116	2.8256	[1]

This experiment took around 8 hours to process in the computer previously described.

4.2 Evaluation

Our system was evaluated against the corpora provided for the *2nd International Competition on Plagiarism Detection*, using the parameters obtained with the training set. These corpora are of the same nature as the ones used to train our system, but with the inclusion of additional novel cases of plagiarism [12]. In addition, they contain a larger number of documents. Concretely, they are composed of $11,148$ source and $15,925$ suspicious documents.

Our system obtained the following results after being applied over the aforementioned corpora: $overall = 0.2222$, $f\text{-}score = 0.3762$, $precision = 0.9308$, $recall = 0.2357$, and $granularity = 2.2332$. As we can see in these values, the recall value was considerably lower to that shown in the training section. We believe this is due to the addition of novel cases of plagiarism in the corpus provided for the *2nd International Competition on Plagiarism Detection*, as our system was tuned to work for the cases present in the corpora of the *1st International Competition on Plagiarism Detection*. This same reason made precision higher as well as granularity. The resulting overall score is considerably lower in the evaluation phase of the experimentation compared to the training because our system is not able to detect the new plagiarism cases introduced in the corpora used for the former. In the end, our system ranked 11^{th} out of 18 participants.

This experiment took around 9 hours to process in the computer previously described.

5 Conclusions and future work

In this paper we have presented an efficient and scalable approach to detect external plagiarisms based on textual similarity comparisons. This has proven to work fairly well as our system's performance has been among the best of the approaches that participated in the *1st International Competition on Plagiarism Detection*. In the second edition, however, results were considerably worse because our system has the disadvantage that it will only behave well for levels of plagiarism obfuscation that are not high, although given the size of the corpora we worked with, applying more complex techniques, such as semantic or syntactic analysis, doesn't seem feasible due to performance constraints. Because of this reason, our system works well with low or medium levels of plagiarism obfuscation, but not with high levels or when the plagiarized document has been translated into a different language.

As future work we would like to focus on smaller corpora in order to be able to apply more complex techniques such as the usage of semantic and syntactic knowledge. For instance, we could use textual entailment recognition techniques, such as the ones presented in [4], to detect plagiarisms that have a high level of obfuscation because they have been rewritten using a different word order or equivalent terms. Furthermore we would also like to apply document language recognition techniques and automatic translators to overcome the problem with translated plagiarized documents.

Acknowledgements

The authors of this paper would like to thank the organizers of the *1st and 2nd International Competitions on Plagiarism Detection* for providing the corpora and evaluation metrics that we have used extensively to develop our system, and for having organized the aforementioned competitions. In addition, we would like to thank Dario Bigongiari and Michael Schueppert for their help in setting up and running the experiments described in this paper.

This research has been partially funded by the Spanish Ministry of Science and Innovation (grant TIN2009-13391-C04-01), the Conselleria d'Educació of the Spanish Generalitat Valenciana (grants PROMETEO/2009/119 and ACOMP/2010/286), and the University of Alicante post-doctoral fellowship program funded by Fundación Caja-Murcia.

References

1. Allen, J.: Submission to the 1st International Competition on Plagiarism Detection. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. San Sebastian, Spain (September 2009)
2. Basile, C., Benedetto, D., Caglioti, E., Cristadoro, G., Esposti, M.D.: A Plagiarism Detection Procedure in Three Steps: Selection, Matches and "Squares". In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 19–23. San Sebastian, Spain (September 2009)
3. Basile, C., Benedetto, D., Caglioti, E., Esposti, M.D.: An example of mathematical authorship attribution. Journal of Mathematical Physics 49, 125211–125230 (2008)

4. Ferrández, Ó., Micol, D., Muñoz, R., Palomar, M.: A Perspective-Based Approach for Solving Textual Entailment Recognition. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 66–71. Prague, Czech Republic (June 2007)
5. Grozea, C., Gehl, C., Popescu, M.: ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 10–18. San Sebastian, Spain (September 2009)
6. Kasprzak, J., Brandejs, M., Křipač, M.: Finding Plagiarism by Evaluating Document Similarities. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 24–28. San Sebastian, Spain (September 2009)
7. Malcolm, J.A., Lane, P.C.R.: Tackling the PAN'09 External Plagiarism Detection Corpus with a Desktop Plagiarism Detector. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 29–33. San Sebastian, Spain (September 2009)
8. Muhr, M., Zechner, M., Kern, R., Granitzer, M.: External and Intrinsic Plagiarism Detection Using Vector Space Models. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 47–55. San Sebastian, Spain (September 2009)
9. Palkovskii, Y.: "Counter plagiarism detection software" and "Counter counter plagarism detection". In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 67–68. San Sebastian, Spain (September 2009)
10. Pereira, R.C., Moreira, V.P., Galante, R.: Intrinsic Plagiarism Detection Using Character n-gram Profiles. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 36–37. San Sebastian, Spain (September 2009)
11. Potthast, M., Stein, B., Eiselt, A., Cedeño, A.B., Rosso, P.: Overview of the 1st International Competition on Plagiarism Detection. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 1–9. San Sebastian, Spain (September 2009)
12. Potthast, M., Stein, B., Eiselt, A., Cedeño, A.B., Rosso, P.: Overview of the 2nd International Competition on Plagiarism Detection. In: Proceedings of the CLEF'10 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. Padua, Italy (September 2010)
13. Scherbinin, V., Butakov, S.: Using Microsoft SQL Server Platform for Plagiarism Detection. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 38–46. San Sebastian, Spain (September 2009)
14. Schleimer, S., Wilkerson, D.S., Aiken, A.: Winnowing: Local Algorithms for Document Fingerprinting. In: Proceedings of ACM SIGMOD International Conference on Management of Data. pp. 76–85. San Diego, USA (June 2003)
15. Spärck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28(1), 11–21 (1972)
16. Vallés Balaguer, E.: Putting Ourselves in SME's Shoes: Automatic Detection of Plagiarism by the WCopyFind tool. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 34–35. San Sebastian, Spain (September 2009)